

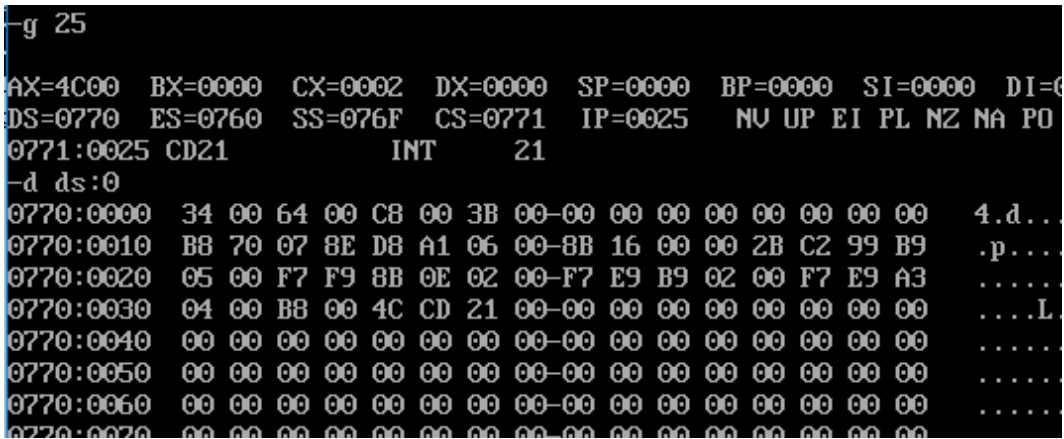
80X86 汇编语言程序设计课内实验报告

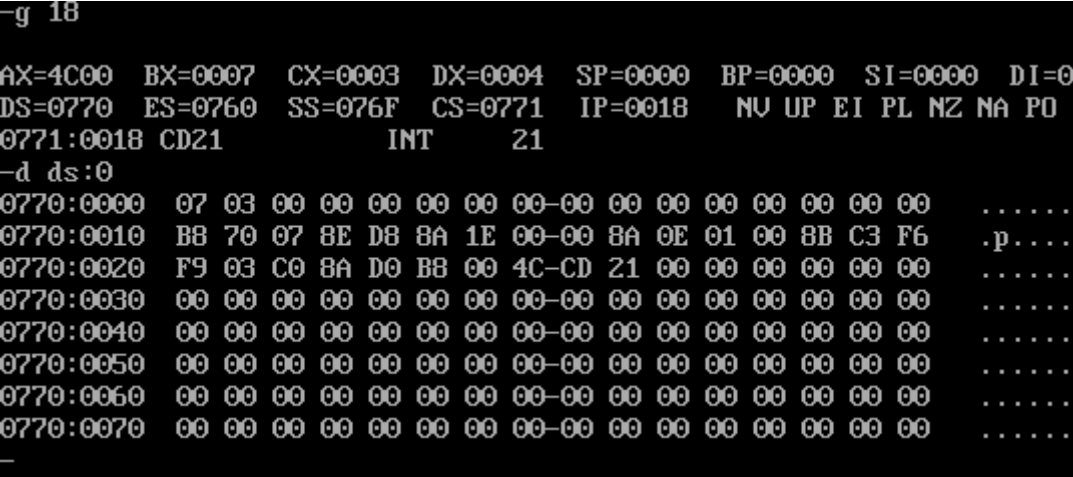
计算机试验班 61

秦佳璐

2140506071

第一次实验

班级	计算机试验班 61	实验时间	2018-03-24
姓名	秦佳璐	学号	2140506071
实验内容概述	实验一： 算术指令程序设计 实验二： 寄存器使用程序设计		
实验步骤	<p>实验一： 算术指令程序设计 完成练习3.17中的第（4）题。设计1个完整的程序，实现执行下列计算功能，其中X、Y、Z、R、W均为存放16位带符号数单元的地址：$Z = ((W - X) / 5 * Y) * 2$</p> <p>设计思路： 将 X、Y、Z、R、W 分别按顺序存储在数据段中，然后按照计算式进行计算。注意：乘法和除法默认与 ax 寄存器中的值进行计算。最后把 ax 寄存器里的最终结果放入 z 存储单元中。</p> <p>实验二： 寄存器使用程序设计 完成练习中3.63题。编制完整程序，要求把BL中的数除以CL中的数，并把其商乘以2，最后结果存入DX寄存器中。</p> <p>设计思路： 1. 按照题目要求，先在 BL 和 CL 中存入数据，做减法 2. 结果存入 ax 再*2 3. 最后结果存入 dx</p>		
实验中遇到的问题及其解决方法	1、问题 本次实验较简单，没有遇到问题。只是刚刚开始编写汇编程序，不熟练，进度较慢		
实验结果	实验一：  <pre> -g 25 AX=4C00 BX=0000 CX=0002 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000 DS=0770 ES=0760 SS=076F CS=0771 IP=0025 NU UP EI PL NZ NA PO 0771:0025 CD21 INT 21 -d ds:0 0770:0000 34 00 64 00 C8 00 3B 00-00 00 00 00 00 00 00 00 4.d... 0770:0010 B8 70 07 BE D8 A1 06 00-8B 16 00 00 2B C2 99 B9 .p... 0770:0020 05 00 F7 F9 8B 0E 02 00-F7 E9 B9 02 00 F7 E9 A3 0770:0030 04 00 B8 00 4C CD 21 00-00 00 00 00 00 00 00 00 L. 0770:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 0770:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 0770:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 0770:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 </pre>		

	<div>实验二：</div> <div></div> <div>年 月 日</div>
建议	
备注	

实验一代码：

DATAS SEGMENT

x dw 34H

y dw 100

z dw 0

w dw 3BH

;此处输入数据段代码

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS, SS:STACKS

START:

mov ax, datas

mov ds, ax

;ax = w - x

MOV AX, w

MOV dx, x

sub ax, dx

;ax = ax / 5

cwd

```

mov cx, 5
idiv cx

;ax = ax * y
mov cx, y
imul cx

;ax = ax * 2
mov cx, 2
imul cx

;z <- ax
mov z, ax

;此处输入代码段代码
mov ax, 4c00H
int 21H
CODES ENDS
END START

```

实验二代码：

```

DATAS SEGMENT
    x db 7
    y db 3
    ;此处输入数据段代码
DATAS ENDS

```

```

;STACKS SEGMENT
    ;此处输入堆栈段代码
;STACKS ENDS

```

```

CODES SEGMENT
    ASSUME CS:CODES, DS:DATAS;, SS:STACKS
START:
    MOV AX, DATAS
    MOV DS, AX

    ;b1 = x; c1 = y
    mov b1, x
    mov c1, y

    ;ax = b1
    mov ax, bx

```

```

;ax = ax / cl = bl / cl
idiv cl

;ax = ax * 2
add ax, ax

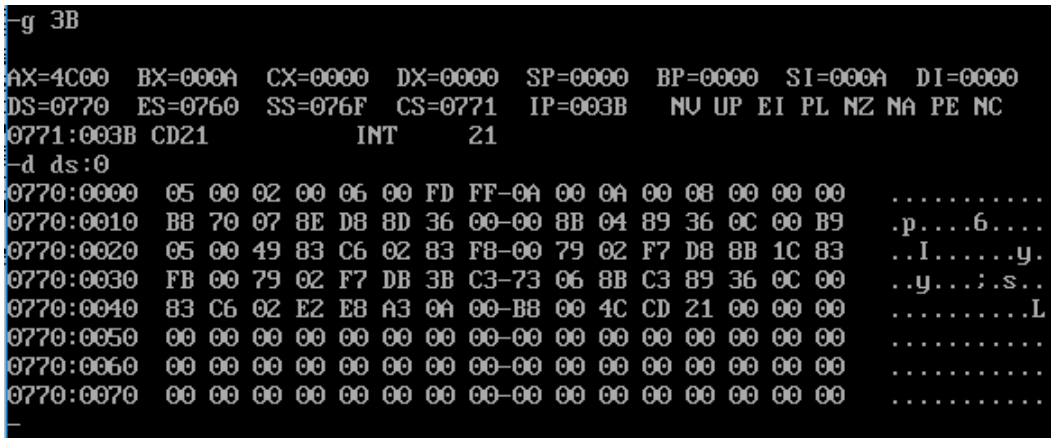
;dx = ax
mov dl, al

;此处输入代码段代码
MOV AX, 4C00H
INT 21H
CODES ENDS
END START

```

第二次实验

班级	计算机试验班 61	实验时间	2018-03-31
姓名	秦佳璐	学号	2140506071
实验内容概述	实验一： 循环程序设计-1 实验二：循环程序设计-2 实验三：分支程序设计		
实验步骤	<p>实验一： 循环程序设计-1 完成练习中 5.15 题。 数据段中已定义了一个有 n 个字数据的数组 M，试编写一程序求出 M 中绝对 值最大的数，把它放在数据段的 M+2n 单元中，并将该数的偏移地址存放在 M+2(n+1)单元中。</p> <p>设计思路：</p> <ol style="list-style-type: none"> 1. 从第一个数开始，记录它的地址和值 2. 设置循环次数，指针 3. 与 0 比大小，大于 0 则转入 abs，小于 0 则取绝对值 4. Abs：找到下一个数，取绝对值，进入 comp 5. Comp：将这个数与下一个数进行比较,始终保持较大数在 ax 寄存器内，然后进入 getnum 6. Getnum：记录该数的值和地址。挪动指针，重新进入循环 <p>实验二：循环程序设计-2 完成练习中 5.19 题。 已知数组 A 包含 15 个互不相等的整数，数组 B 包含 20 个互不相等的整数。 请编制程序，把既在 A 中又在 B 中出现的整数存放于数组 C 中。</p> <p>设计思路：</p>		

	<ol style="list-style-type: none"> 1. 二重循环：每重循环遍历一组数据 2. Loop1: cx 循环次数要压栈，不影响第二重循环，将被比较的数存入 ax 中 3. Loop2: 直接与 ax 中的数比较，如果不相等则进入 continue，如果相等的时候 move 进新的数组，指针下移 4. Continue: 指针下移，循环 loop2，再次进入 loop 时，需要弹出 <p>实验三：分支程序设计</p> <p>设计一程序，接收从键盘输入的数字(1~5)，并对每个数字的输入次数进行计数，计数结果存放在以 COUNT 为首地址的数组中。(最好能将计数结果打印出来)。</p> <p>设计思路：</p> <ol style="list-style-type: none"> 1. 五种选择分别跳入五个不同标号 2. 每个子程序：计数+1，回到 menu 3. Exit: 打印每个的值
实验中遇到的问题及其解决方法	<ol style="list-style-type: none"> 1、问题 <ul style="list-style-type: none"> ■ 问题描述：如何打印在屏幕上 ■ 问题是在实验 X 中遇到，解决的思路是调用 int 21h 2 号功能 ■ 解决问题的步骤是选取 AL=30 在屏幕上显示，然后返回
实验结果	<p>实验一：</p>  <p>实验二：</p>

	<div><div><div><div><div><div>-g 33</div></div></div><div><div><div>AX=4C1C</div><div>BX=0014</div><div>CX=0000</div><div>DX=0000</div><div>SP=0000</div><div>BP=0000</div><div>SI=001E</div><div>DI=0028</div></div><div><div>DS=0770</div><div>ES=0760</div><div>SS=076F</div><div>CS=0777</div><div>IP=0033</div><div>NU UP EI PL NZ NA PE NC</div></div></div></div><div><div><div>0777:0033</div><div>CD21</div><div>INT</div><div>21</div></div></div><div><div><div>-d ds:0</div></div><div><div><div>0770:0000</div><div>00 00 02 00 04 00 06 00-08 00 0A 00 0C 00 0E 00</div><div>.....</div></div><div><div>0770:0010</div><div>10 00 12 00 14 00 16 00-18 00 1A 00 1C 00 00 00</div><div>.....</div></div><div><div>0770:0020</div><div>01 00 02 00 03 00 04 00-05 00 06 00 07 00 08 00</div><div>.....</div></div><div><div>0770:0030</div><div>09 00 0A 00 0B 00 0C 00-0D 00 0E 00 0F 00 10 00</div><div>.....</div></div><div><div>0770:0040</div><div>11 00 12 00 13 00 00 00-02 00 04 00 06 00 08 00</div><div>.....</div></div><div><div>0770:0050</div><div>0A 00 0C 00 0E 00 10 00-12 00 00 00 00 00 00</div><div>.....</div></div><div><div>0770:0060</div><div>00 00 00 00 00 00 00 00-00 00 00 00 00 00 00</div><div>.....</div></div><div><div>0770:0070</div><div>B8 70 07 8E D8 BE 00 00-BB 00 00 B9 0F 00 51 B9</div><div>.p.....</div></div></div></div><div><div><div>-</div></div><div><div><div>-</div></div></div></div><div>实验三:</div><div><div><div><div><div>BOX</div><div>DOSBox 0.74, Cpu speed: 30</div></div></div><div><div><div>22445</div><div>020210_</div></div></div></div></div></div><div>年 月 日</div></div>
建议	
备注	

实验一代码:

DATA1 SEGMENT

M dw 5,2,6,-3,10

data dw ?;存数据

adr dw ?;存地址

DATA1 ENDS

CODES SEGMENT

ASSUME CS:CODES, DS:DATA1

START:

MOV AX, DATA1

MOV DS, AX

lea si, M;取地址

mov ax, [si];取值

mov adr, si

mov cx, 5

dec cx

add si, 2

```
    cmp ax, 0
    jns abs
    neg ax
```

abs:

```
    mov bx, [si]
    cmp bx, 0
    JNS comp
    neg bx
```

comp:

```
    cmp ax, bx
    JAE getnum
    mov ax, bx
    mov adr, si
```

getnum:

```
    add si, 2
    loop abs
    mov data, ax
```

```
    MOV AX, 4C00H
    INT 21H
```

```
CODES ENDS
    END START
```

实验二代码:

DATAS SEGMENT

```
    A dw 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28
    B dw 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
    cc dw 15 dup(?)
```

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

```
    ASSUME CS:CODES, DS:DATAS, SS:STACKS
```

START:

```
    MOV AX, DATAS
    MOV DS, AX
```



```
mov si, 0
mov bx, 0
mov cx, 15
```

```
loop1:
    push cx
    mov cx, 20
    mov di, 0
    mov ax, A[si]
```

```
loop2:
    cmp B[di], ax
    JNZ continue
    mov cc[bx], ax;相等的时候move
    add bx, 2
```

```
continue:
    add di, 2
    loop loop2
    add si, 2
    pop cx
    loop loop1
```

```
MOV AH, 4CH
INT 21H
CODES ENDS
END START
```

实验三代码:

```
DATAS SEGMENT
    num db 5 dup(0)
DATAS ENDS
```

```
STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS
```

```
CODES SEGMENT
    ASSUME CS:CODES, DS:DATAS, SS:STACKS
START:
    MOV AX, DATAS
    MOV DS, AX
```

menu:

```
mov ah, 01h
int 21h
cmp al, '1'
jz n1
cmp al, '2'
jz n2
cmp al, '3'
jz n3
cmp al, '4'
jz n4
cmp al, '5'
jz n5
jmp exit
```

```
n1:
    mov cl, num
    inc cl
    mov num[0], cl
    jmp menu
n2:
    MOV BX, 1
    MOV CL, NUM[BX]
    inc cl
    mov num[bx], cl
    jmp menu
n3:
    MOV BX, 2
    MOV CL, NUM[BX]
    inc cl
    mov num[bx], cl
    jmp menu
n4:
    MOV BX, 3
    MOV CL, NUM[BX]
    inc cl
    mov num[bx], cl
    jmp menu
n5:
    MOV BX, 4
    MOV CL, NUM[BX]
    inc cl
    mov num[bx], cl
    jmp menu
```

```

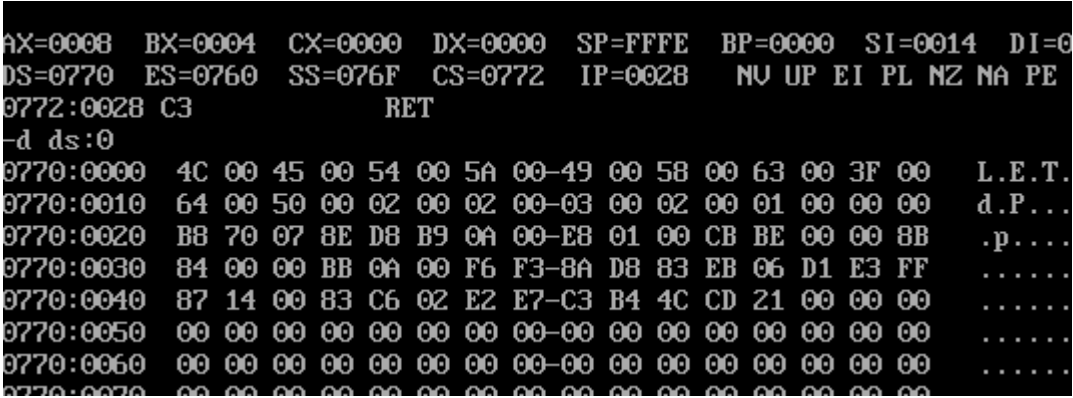
exit:
    mov dl,num[0]
    add dl,30h
    mov ah,02
    int 21h
    mov dl,num[1]
    add dl,30h
    mov ah,02
    int 21h
    mov dl,num[2]
    add dl,30h
    mov ah,02
    int 21h
    mov dl,num[3]
    add dl,30h
    mov ah,02
    int 21h
    mov dl,num[4]
    add dl,30h
    mov ah,02
    int 21h
    mov dl,num[5]
    add dl,30h
    mov ah,02
    int 21h

    MOV AH,01H
    INT 21H
CODES ENDS
    END START

```

第三次实验

班级	计算机试验班 61	实验时间	2018-04-20
姓名	秦佳璐	学号	2140506071
实验内容概述	实验一：子程序设计 实验二：递归子程序设计 实验三：高级汇编语言技术		
实验步骤	实验一：子程序设计 完成练习中 6.9 题。设有 10 个学生的成绩分别是 76, 69, 84, 90, 73, 88, 99, 63, 100 和 80 分。试编制一个子程序统计 60~69 分, 70~79 分, 80~89 分, 90~99 分和 100 分的人数并分别 存放到 S6, S7,		

	<p>S8, S9 和 S10 单元中。</p> <p>设计思路：</p> <ol style="list-style-type: none">1. 循环次数cx设为10，设置指针位置2. 将分数除以10，只看十位进行分组。3. 将对应组别计数+1 <p>实验二：递归子程序设计</p> <p>完成练习中 6.13 题。 给定一个正数 $N \geq 1$ 存放在 NUM 单元中，试编制一递归子程序计算 FIB (N)，并 将结果存入 RESULT 单元中。 Fibonacci 数的定义如下： FIB (1) = 1 FIB (2) = 1 FIB (n) = FIB (n-2) FIB (n-1) $n > 2$</p> <p>设计思路：</p> <ol style="list-style-type: none">1. 主程序为：设置项数存入ax，然后运行fib递归。最后结果存入result单元2. Fib递归：项数1和2的时候直接赋值。项数大于2的时候，减1并再次进入fib递归。3. Ender: pop栈内内容，记录最终结果。 <p>实验三：高级汇编语言技术</p> <p>完成练习中 7.11 题。 试编写一段程序，完成以下功能：若给定名为 X 的字符串长度大于 5 时，下列指 令将汇编 10 次。 ADD AX, AX (要求：使用条件汇编，编译过程中生成 LST 文件，查看宏展开情况)</p> <p>设计思路：</p> <ol style="list-style-type: none">1. 设置两个串，将两个串的首地址相减，与长度 5 进行比较2. 大于 5 就重复 10 次，小于 5 则不重复
实验中遇到的问题及其解决方法	<p>1、问题</p> <ul style="list-style-type: none">■ 问题描述：fib 递归里，无论如何最后数据段里没有值■ 问题是在实验二中遇到，解决的思路是重新查找程序，整理思路，寻找应在哪里把值最后存入 result 中■ 解决问题的步骤是细心查验
实验结果	<p>实验一：</p> 

实验二：

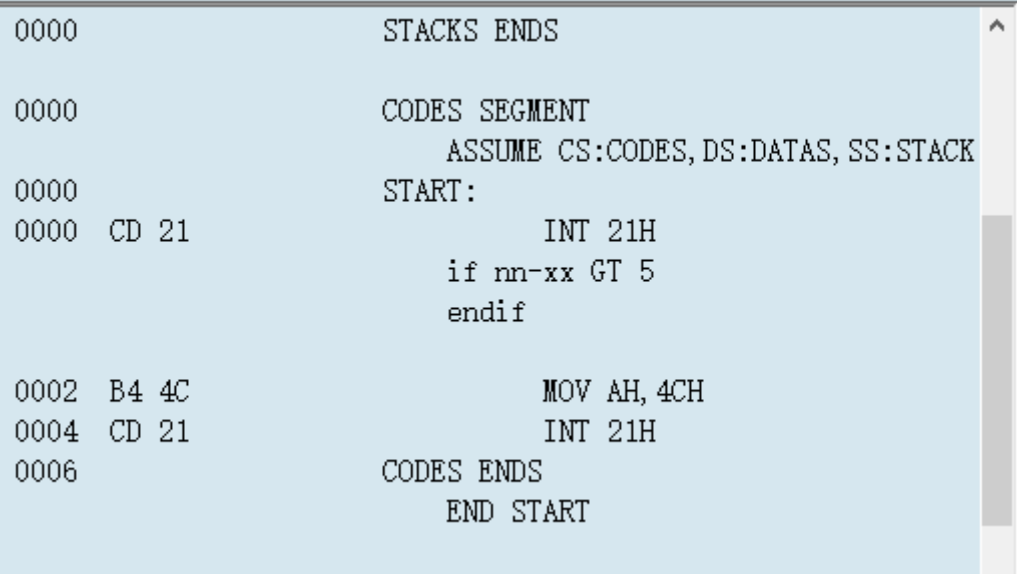
```
-g 11
AX=4C0E BX=0179 CX=004A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0011  NU UP EI PL NZ NA P
0771:0011 CD21          INT     21
-d ds:0
0770:0000 0E 00 79 01 00 00 00 00-00 00 00 00 00 00 00 00  ..y.
0770:0010 BB 70 07 8E D8 A1 00 00-E8 08 00 89 1E 02 00 B4  .p.
0770:0020 4C CD 21 50 51 83 F8 02-7E 13 83 F8 01 7E 11 48  L.!F
0770:0030 E8 F0 FF 8B CB 48 E8 EA-FF 03 D9 EB 06 BB 01 00  ...
0770:0040 BB 01 00 59 58 89 1E 02-00 C3 00 00 00 00 00 00  ...Y
0770:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ...
0770:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ...
0770:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ...
-
```

实验三：

```
01 DATAS SEGMENT
02     ;此处输入数据段代码
03     xx db 'ssasdh'
04     nn db ' '
05 DATAS ENDS
06
11 nn-xx 01 0
    rept 10
        add ax, ax
    endm
0002 03 C0          1          add ax, ax
0004 03 C0          1          add ax, ax
0006 03 C0          1          add ax, ax
0008 03 C0          1          add ax, ax
000A 03 C0          1          add ax, ax
000C 03 C0          1          add ax, ax
000E 03 C0          1          add ax, ax
0010 03 C0          1          add ax, ax
0012 03 C0          1          add ax, ax
0014 03 C0          1          add ax, ax
    endif

0016 B4 4C          MOV AH, 4CH
0018 CD 21          INT 21H
001A                CODES ENDS
                END START

;此处输入数据段代码
xx db 'asdh'
nn db ' '
DATAS ENDS
```

	
建议	年 月 日
备注	

实验一代码：

DATAS **SEGMENT**

;此处输入数据段代码

score dw 76, 69, 84, 90, 73, 88, 99, 63, 100, 80

s6 dw 0

s7 dw 0

s8 dw 0

s9 dw 0

s10 dw 0

DATAS **ENDS**

STACKS **SEGMENT**

;此处输入堆栈段代码

STACKS **ENDS**

CODES **SEGMENT**

ASSUME CS:CODES, DS:DATAS, SS:STACKS

START:

main proc far

MOV AX, DATAS

MOV DS, AX

;此处输入代码段代码

begin:

mov cx, 10

```

        call count
        ret
main endp

count proc near
    mov si,0
next:
    mov ax,score[si]
    mov bx,10
    div bl
    mov bl,al
    sub bx,6
    sal bx,1
    inc s6[bx]
    add si,2
    loop next
    ret
count endp

```

```

        MOV AH,4CH
        INT 21H
CODES ENDS
        END START

```

实验二代码:

```

DATAS SEGMENT
    ;此处输入数据段代码
    num dw 14
    result dw ?
DATAS ENDS

```

```

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

```

```

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码

    mov ax,num

```

```

    call fib
    mov result,bx
    MOV AH,4CH
    INT 21H

fib proc near
    push ax
    push cx

    cmp ax,2
    jle less_equal_2
    cmp ax,1
    jle less_equal_1

    dec ax
    call fib
    mov cx,bx
    dec ax
    call fib
    add bx,cx

    jmp ender

less_equal_2:
    mov bx,1
less_equal_1:
    mov bx,1

ender:
    pop cx
    pop ax
    mov result,bx
    ret

fib endp

CODES ENDS
    END START

```

实验三代码：

```

DATAS SEGMENT
    ;此处输入数据段代码
    xx db ' asdh'
    nn db ' '

```



```

DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

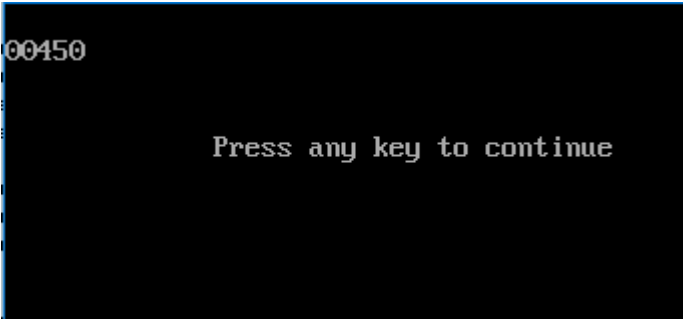
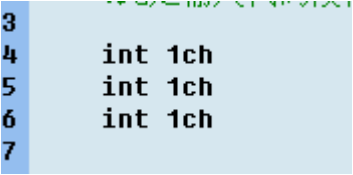
CODES SEGMENT
    ASSUME CS:CODES, DS:DATAS, SS:STACKS
START:
    INT 21H
    if nn-xx GT 5
        rept 10
            add ax, ax
        endm
    endif

    MOV AH, 4CH
    INT 21H
CODES ENDS
    END START

```

第四次实验：

班级	计算机试验班 61	实验时间	2018-05-02
姓名	秦佳璐	学号	2140506071
实验内容概述	实验一： 输入输出程序设计 实验二： 中断程序设计 实验三： BIOS 和 DOS 中断		
实验步骤	<p>实验一： 输入输出程序设计 完成练习中 8.14 题。 试编制一程序，要求测出任意程序的运行时间，并把结果在屏幕打印出来。</p> <p>设计思路：</p> <ol style="list-style-type: none"> 1. 基本思路是记录两次系统时间，相减并打印 2. DOS的2C功能是获取系统时间。其中秒和1/100秒分别存储在dh和dl 3. 存储模式是将 秒*100+百分之一秒，单位为1/100秒 4. 将两次的系统时间相减，一步步处以10，按10进制输出 <p>实验二： 中断程序设计 编写程序，使类型 1CH 的中断向量指向中断处理程序 COUNT，COUNT 统计 1CH 中 断次数并存入单元 M 中。</p> <p>设计思路：</p> <ol style="list-style-type: none"> 1. 取中断类型号1C 		

	<div>2. 取中断向量</div> <div>3. 重新设置中断向量</div> <div>4. 转入中断程序，中断多次并记录次数</div> <div>5. 返回主程序</div> <div><div>实验三： BIOS 和 DOS 中断</div><div>完成练习中 9.2 题。 编写一个程序，接收从键盘输入的 10 个十进制数字，输入回车符则停止输入， 然后将这些数字加密后（用 XLAT 指令变换）存入内存缓冲区 BUFFER。加密表 为： 输入数字： 0， 1， 2， 3， 4， 5， 6， 7， 8， 9 密码数字： 7， 5， 9， 1， 3， 6， 8， 0， 2， 4</div><div>设计思路：</div><div>1. 从键盘接受输入，回车即停止输入，同时用 xlat 加密</div></div>
实验中遇到的问题及其解决方法	<div>1、问题</div> <div><div>■ 问题描述：每 18.2s 自动进行一次中断，记录中断次数中断次数即可的算法，程序过于复杂</div><div>■ 问题是在实验一中遇到，解决的思路是可以取两次系统时间，直接相减。</div><div>■ 解决问题的步骤是去两次系统时间，将秒和百分之一秒合在一起。相减并打印。</div></div>
实验结果	<div>实验一：</div> <div></div> <div>实验二：</div> <div></div>

	<div><div><div><div><div><div>-g 26</div></div></div><div><div><div>AX=0770</div> <div>BX=1260</div> <div>CX=0047</div> <div>DX=0032</div> <div>SP=FFF8</div> <div>BP=0000</div> <div>SI=0000</div> <div>DI=0</div></div><div><div><div>DS=0770</div> <div>ES=F000</div> <div>SS=076F</div> <div>CS=0771</div> <div>IP=0026</div> <div>NU UP EI PL NZ NA PO</div></div><div>0771:0026 5A POP DX</div></div></div><div><div><div>-d ds:0</div></div></div><div><div><div>0770:0000</div> 03 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00</div><div>0770:0010 1E B8 00 00 50 B0 1C B4-35 CD 21 06 53 B8 71 07P.</div><div>0770:0020 8E D8 BA 32 00 B0 1C B4-25 CD 21 B8 70 07 8E D8 ...2..</div><div>0770:0030 CD 1C CD 1C CD 1C 5A 1F-B0 1C B4 25 CD 21 B4 4C</div><div>0770:0040 CD 21 FF 06 00 00 CF 00-00 00 00 00 00 00 00 00 !....</div><div>0770:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00</div><div>0770:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00</div></div></div></div><div>实验三:</div><div><div><div><div><div><div>0772:0040</div> 0000</div><div>ADD</div><div> [BX+SI],AL</div></div></div></div></div><div><div><div>-g 26</div></div></div><div>9987654321</div><div><div><div><div><div><div>AX=4C0D</div> <div>BX=000A</div> <div>CX=003E</div> <div>DX=0000</div> <div>SP=0000</div> <div>BP=0000</div> <div>SI=000A</div> <div>DI=0000</div></div><div><div><div>DS=0770</div> <div>ES=0760</div> <div>SS=076F</div> <div>CS=0772</div> <div>IP=0026</div> <div>NU UP EI PL ZR NA PE NC</div></div><div>0772:0026 CD21 INT 21</div></div></div><div><div><div>-d ds:0</div></div></div><div><div><div>0770:0000</div> 04 04 02 00 08 06 03 01-09 05 07 05 09 01 03 06</div><div>0770:0010 08 00 02 04 00 00 00 00-00 00 00 00 00 00 00 00</div><div>0770:0020 B8 70 07 8E D8 BE 00 00-8D 1E 0A 00 B4 01 CD 21 .p.....</div><div>0770:0030 3C 0D 74 10 2C 30 72 F4-3C 09 77 F0 D7 88 84 00 <.t.,0r.<.u</div><div>0770:0040 00 46 E2 E8 B4 4C CD 21-00 00 00 00 00 00 00 00 .F...L.!...</div><div>0770:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00</div><div>0770:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00</div><div>0770:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00</div></div></div></div><div>年 月 日</div></div></div>
建议	
备注	

实验一代码:

DATAS SEGMENT

 ;此处输入数据段代码

 time1 dw 0

 time2 dw 0

DATAS ENDS

STACKS SEGMENT

 ;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS, SS:STACKS

START:

MOV AX, DATAS

```
MOV DS, AX  
;此处输入代码段代码
```

```
;获取系统时间
```

```
mov ah, 2ch  
int 21h
```

```
;1/100秒到bx
```

```
mov bx, 0  
mov cl, dl  
mov ch, 0  
add bx, cx
```

```
;秒*100到bx, 与前面错位
```

```
mov al, dh  
mov dh, 100  
mul dh  
add bx, ax
```

```
mov time1, bx
```

```
; 中断
```

```
mov ah, 01h  
int 21h
```

```
;获取系统时间
```

```
mov ah, 2ch  
int 21h
```

```
;处理时间
```

```
mov bx, 0  
mov cl, dl  
mov ch, 0  
add bx, cx
```

```
mov al, dh  
mov dh, 100  
mul dh  
add bx, ax
```

```
mov time2, bx
```

```
;时间相减
```

```
mov ax, time2
```

```

mov bx, time1
sub ax, bx

mov cx, 5      ;输出位数
mov bl, 10     ;10进制输出

```

```

print:
    div bl      ;余数放在ah, 商放在al中

```

```

    mov dl, ah
    add dl, 30h
    push dx
    mov bh, al
    mov al, bh
    mov ah, 0
    loop print

```

```

    mov cx, 5

```

```

exit:
    pop dx
    mov ah, 2h
    int 21h
    loop exit

```

```

MOV AH, 4CH
INT 21H

```

```

CODES ENDS
END START

```

实验二代码:

```

DATAS SEGMENT
    ;此处输入数据段代码
    m dw 0
DATAS ENDS

```

```

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

```

```

CODES SEGMENT
    ASSUME CS:CODES, DS:DATAS, SS:STACKS
START:
    push ds

```

```
mov ax, 0
push ax
```

```
mov al, 1ch
mov ah, 35h
int 21h
```

```
push es
push bx
mov ax, seg count
mov ds, ax
mov dx, offset count
mov al, 1ch
mov ah, 25h
int 21h
```

```
MOV AX, DATAS
MOV DS, AX
;此处输入代码段代码
```

```
int 1ch
int 1ch
int 1ch
```

```
pop dx
pop ds
mov al, 1ch
mov ah, 25h
int 21h
```

```
MOV AH, 4CH
INT 21H
```

```
count proc near
    inc m
    iret
count endp
```

```
CODES ENDS
END START
```

实验三代码：

```
DATAS SEGMENT
```

```

;此处输入数据段代码
buffer db 10 dup(?)
passw db 7,5,9,1,3,6,8,0,2,4
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES, DS:DATAS, SS:STACKS
START:
main proc far
    MOV AX, DATAS
    MOV DS, AX
;此处输入代码段代码

    mov si, 0
    lea bx, passw

loop1:
    mov ah, 01h
    int 21h
    cmp al, 0dh
    jz exit
    sub al, 48
    jb loop1
    cmp al, 09h
    ja loop1
    xlat
    mov buffer[si], al
    inc si
    loop loop1

exit:
    MOV AH, 4CH
    INT 21H
main endp
CODES ENDS
    END START

```